* Describe what happens with the following instruction executes:
* J Loop

Jumps to the top of the loop

* LUI $t0, 0xffff

Load Upper Immediate.

The immediate value (1111 1111 1111 1111 or 65535) is shifted left 16 bits and stored in the $t0 register.

* Diagram the “stair step” instruction execution for the following instructions. Assume no forwarding:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | |
| LW $t0, 0($s1) | | IF | ID | EX | MEM | WB |  |  |  |  |  |  |  |  |  |  | |
| ADDI $t3, $t0, 1 | |  |  |  | IF | ID | EX | MEM | WB |  |  |  |  |  |  |  | |
| SUB $t2, $t4, $t5 | |  |  |  |  | IF | ID | EX | MEM | WB |  |  |  |  |  |  | |
| SW $t3, 0($s1) |  | |  |  |  |  |  | IF | ID | EX | MEM | WB |  |  |  |  |

--This is wrong, ADDI, SUB, and SW should all be shifted left one. Below

* Diagram the “stair step” instruction execution for the same instructions, but this time, assume forwarding

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| LW $t0, 0($s1) | IF | ID | EX | MEM | WB |  |  |  |  |  |  |  |  |  |  |
| ADDI $t3, $t0, 1 |  |  |  | IF | ID | EX | MEM | WB |  |  |  |  |  |  |  |
| SUB $t2, $t4, $t5 |  |  |  |  | IF | ID | EX | MEM | WB |  |  |  |  |  |  |
| SW $t3, 0($s1) |  |  |  |  |  | IF | ID | EX | MEM | WB |  |  |  |  |  |

* What is the percent speedup of the execution in problem 3 compared with problem 2?

2200ps vs 2000ps

2000ps vs 1800ps

\* 1 million

1m \* 2000ps + 2200ps = 2000002200ps

1m \* 1800ps + 2000ps = 1800002000ps

2000002200ps / 1800002000ps = 1.111111099 or ~11% faster